PERGAMON

Pattern Recognition 36 (2003) 245–257

PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY

www.elsevier.com/locate/patcog

Unsupervised face recognition by associative chaining

Bisser Raytchev*, Hiroshi Murase

NTT Communication Science Laboratories, NTT Corporation, 3-1 Morinosato Wakamiya, Atsugi-shi, Kanagawa 243-0198, Japan

Received 27 July 2001; accepted 12 March 2002

Abstract

We propose a novel method for unsupervised face recognition from time-varying sequences of face images obtained in real-world environments. The method utilizes the higher level of sensory variation contained in the input image sequences to autonomously organize the data in an incrementally built graph structure, without relying on category-specific information provided in advance. This is achieved by "chaining" together similar views across the spatio-temporal representations of the face sequences in image space by two types of connecting edges depending on local measures of similarity. Experiments with real-world data gathered over a period of several months and including both frontal and side-view faces from 17 different subjects were used to test the method, achieving correct self-organization rate of 88.6%. The proposed method can be used in video surveillance systems or for content-based information retrieval. © 2002 Pattern Recognition Society. Published by Elsevier Science Ltd. All rights reserved.

Keywords: Face recognition; Unsupervised incremental learning; Time-varying image sequences; Video surveillance

1. Introduction

In recent years automated face recognition has attracted a lot of attention. This seems to be motivated not only by scientific curiosity, but also by the numerous potential applications stemming from the fact that faces represent natural interfaces for humans, and face recognition is central to human communication. However, in spite of the extensive research conducted in this area during the last several decades (see Refs. [1–4] for surveys), face recognition still remains a domain in which humans significantly outperform computers, especially in real-time, unconstrained and unpredictable environments. Here we argue that some of the reasons for this situation, together with hints for the answers, might be found by investigating some of the discrepancies between the way humans learn faces and the way most computer-based face recognition procedures operate:

(a) Humans learn by interacting directly with the sensory input from their environment. Category labels, like human names in the case of face recognition, are not essential for discrimination in the learning process and are used just for convenience *after* the faces have already been learnt, based on the internal characteristics of the sensory input itself (*unsupervised learning*), rather than on any category-specific information accompanying it in a supervised manner. This is in contrast to the way most computer-based face recognition procedures operate. Computers are usually provided with input, which has been segmented and classified in advance (*supervised learning*) by human teachers, and as a result of this might be biased by their limited understanding of the complex real-world environment;

(b) Biological learning is *incremental* in nature, i.e. new categories can be learnt and added to those already in existence, without the need to "relearn" everything anew, or to represent the new categories with a restricted pre-defined set of features, which in the case of computer learning are either designed by humans or automatically selected to represent the *available* data in some optimal way. The number of different categories to be learnt is not fixed and known in advance—the learning system must be open for new additions at any time;

(c) Automatic face recognition is difficult because different people's faces observed in the same conditions

* Corresponding author. Tel.: +81-46-240-3539; fax: +81-46-240-4708.

E-mail address: bisser@eye.brl.ntt.co.jp (B. Raytchev).

PII: S0031-3203(02)00068-7

(illumination, view angle, size, etc.) look more similar to each other than the same person's face observed in different conditions (e.g. in frontal and side view; under extreme illumination conditions; occluded, etc.). One approach to solve this problem is to find features invariant under different conditions, but this has proven to be difficult. It might be possible that biological systems use a different approach—to learn from time-sequential input, in the form of temporally-constrained continuous sensory streams, containing the whole spectrum of variations in illumination, viewing angles and object sizes, which everyday life provides. Again, in contrast to this, computers typically are trained with *few isolated* samples from a large set of different face categories, taken in restricted environmental conditions.

Although some researchers have already pointed out the need for incremental and unsupervised self-organization of the internal state of the learning system Refs. [5–7], see also Ref. [8] for a relevant discussion on the differences between human and machine learning and the need for "more cognitive learning"), or use of time-sequential data [9], a method for face recognition which takes into consideration all of the concerns mentioned above and performs reasonably well on real-world data has not been demonstrated yet, to our knowledge.

In this paper we propose a new method for unsupervised face recognition from video sequences of time-varying facial images, inspired by observations (a)–(c) above. The method utilizes the higher level of sensory variation contained in the input image sequences to autonomously organize the data in an incrementally built graph structure, without relying on category-specific information provided in advance. This is achieved by "chaining" together associations (similar views) across the spatio-temporal representations of the face sequences in image space, by two types of connecting edges depending on local measures of similarity. Several experiments, using data obtained in real-world conditions, were conducted in order to evaluate the performance of the proposed method, and encouraging results were observed. Expected areas of application of this method include visitor identification in surveillance systems, content-based face retrieval/annotation in multimedia applications, etc.

2. Learning by associative chaining

The purpose of the learning algorithm introduced here is to group a set of unlabeled face image sequences, which could be pre-stored as a database (*batch mode*), or obtained in a sequential manner in the order they become available from an input device (*incremental mode*). As already mentioned, this has to be done without using any category information provided in advance, i.e. some clustering technique (e.g. see Refs. [10–12]) has to be utilized. Our task is further complicated by the following requirements: (a) generally, the number of face categories is not known in advance and newly available categories have to be accounted for in a non-destructive manner; (b) the different categories are not represented uniformly, some might be under-represented and some over-represented; (c) in sample space, the face sequences for the different face categories form non-linear manifolds with complex structure, for which intra-class distances can take higher values than inter-class distances. The above-mentioned characteristics of the problem preclude the possibility of using some of the popular clustering approaches, and this has motivated us to propose the current method.

The clustering algorithm proposed here, which we call associative chaining (AC), has been implemented as the core part of a fully automatic system for face recognition from image sequences, which operates in several stages. The role of the first, *preprocessing stage,* is to automatically extract and normalize the face area of the subjects appearing in video sequences containing dynamic scenes of moving people, and provide them as an input to the next stage, in the form of time-segmented face image sequences. It is assumed that the face of each individual appearing in a scene can be tracked online, and as a result of the tracking and face-extraction process (a simplified scenario described in Section 2.1), a separate face-only image sequence can be produced for the time interval from each individual's appearance in the scene to his/her disappearance. In the *learning stage*, the associative chaining algorithm is run on the accumulated face-only image sequences in order to organize them into category groups, i.e. to partition the input sample space into face clusters, without using category-specific information. The learning process can operate either in *batch mode* or in *incremental mode*, depending on the requirements of the concrete application. First, the batch version of the AC algorithm will be introduced in Sections 2.2 and 2.3, from which will be derived the incremental version in Section 2.4. Online face *recognition/verification*, which in the frame of our system can be viewed as instances of *incremental node addition*, will also be treated in Section 2.4.

2.1. Preprocessing of the input

Although the concrete implementation of this part of the system is not essential for the operation of the learning algorithm, which is our major concern in this paper, still some procedure for automatic extraction of faces (or, alternatively, other objects of interest) from image sequences is needed, to provide the necessary input data. All that is required from the preprocessing module is to obtain somehow image sequences of the moving objects of interest and to guarantee that each separate image sequence belongs to one and the same category, i.e. objects from different categories would not appear in the same sequence. This is a reasonable assumption, having in mind that in the 3D world we occupy, it is unlikely that a certain person's face would suddenly turn

Fig. 1. An example of the original face image sequence (temporally subsampled) together with the corresponding normalized face-only sequence extracted from it.

into someone else's face. The input data which we used to test our system were obtained under the following experimental setting. A video camera was fixed in a constant position, continuously monitoring the scene in front of it. Each of the subjects enters the scene (one at a time), walks towards the camera and finally exits the scene passing near the camera (see Fig. 1 for a typical image sequence). The following algorithm was used to extract face-only image sequences from the continuous input stream. After the detection of a subject entering the scene, and until that subject exits the scene, the images taken by the camera are subtracted from the most recently saved background-only image (which is continuously updated in the absence of foreground action), and thresholded to obtain the binary images $B(x, y)$, where x and y are image coordinates. A multi-resolution image pyramid $B^r(x, y)$ is formed from $B(x, y)$, and the binary silhouettes $S^r(x, y)$ of the subject are extracted at each resolution level r of the pyramid. Next, the x- and y-histograms $H^r(x)$ and $H^r(y)$ are calculated from $S^r(x, y)$, analyzed in order to find the shoulder line of the silhouette, and then the face area frame coordinates are determined from the assumption that the head is the blob above the shoulder line. After the face coordinates are calculated at each level of the resolution pyramid, their median value is taken as the final result. Using this information, the face area is extracted from the original image and normalized into an 18×22 pixels face-only image. The faces obtained for the time period between the subject's entering and leaving the scene are stacked into a single face-only image sequence, which will be input to the next stage of the system for learning it.

Obviously, this simple face detection procedure would not work under more complicated scenarios in which the assumptions above are not satisfied (e.g. multiple users partly occluding each other, non-static backgrounds, etc.), but it proved sufficient in our case to collect the input data necessary to test the learning algorithm. The resulting face extraction was not as precise as would have been achieved by manual face extraction (an unattractive alternative, having in mind that tens of thousands of faces had to be processed for the experiments reported in Section 3), and occasionally produced faces which were misaligned, wrongly cropped and of slightly different scale, as can be seen by careful inspection of the face images in Figs. 1–3, for example, but as a whole the results seemed acceptable, and additionally had the effect of testing the algorithm in the presence of noisy data. Face detection/tracking is a very rapidly developing area (see, for example Refs. [13–16]), and we believe that employing a more reliable preprocessor than the one we have used would lead to better performance of the learning module, too, but still a perfect face detection and tracking under real-world conditions might not be easily achievable, and therefore the learning algorithm should be able to tolerate significant amounts of noise.

2.2. Minimal spanning tree (MST) formation

In order to organize the set of face image sequences into clusters, first it is necessary to define a suitable measure for the distance between two image sequences. Let $F^{(a)}(i, j, t)$ and $F^{(b)}(i, j, t)$ be two face image sequences, where a and b are sequence indexes ($a, b : 1 \ldots N$), i and j are image coordinates, and t is image frame number (different sequences might have different length). For all available face image sequences $F^{(n)}(i, j, t)$ compute the dissimilarity (distance) matrices $\boldsymbol{M}^{(a,b)}$ whose elements $m^{(a,b)}\{x, y\}$ represent the dissimilarity (distance) between the xth face and yth face, respectively, in $F^{(a)}(i, j, x)$ and $F^{(b)}(i, j, y)$:

$$m^{(a,b)}(x, y) = dist\langle F^{(a)}(i, j, x),\ F^{(b)}(i, j, y)\rangle = \sum_{i,j} T_\delta\{|F^{(a)}(i, j, x) - F^{(b)}(i, j, y)|\}, \quad (1)$$

where $T_\delta(x)$ is a threshold function with suitable threshold parameter δ. In the face distance measure (1), which was proposed in Ref. [17], the dissimilarity between two grayscale face images is computed by subtracting them from each other and counting the number of pixel positions which differ by more than δ. Although alternative measures for face dissimilarity might be used, we have chosen Eq. (1)

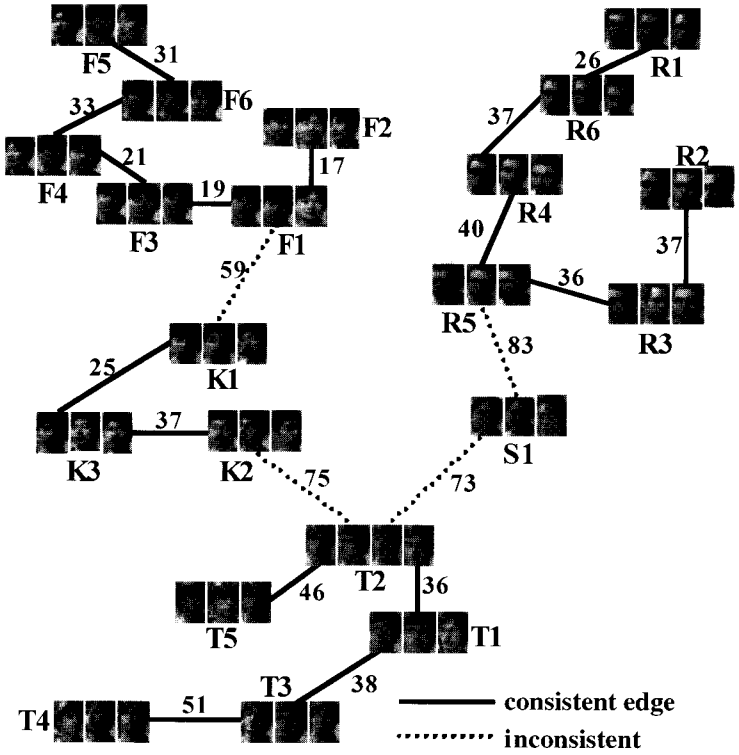


Fig. 2. An example of a graph $\mathbf{G}(\mathbf{C}, \mathbf{E}, \Gamma, \rho)$ obtained during the MST formation stage. Different letters are used for the nodes corresponding to face image sequences from different categories, and the numbers after the letters represent the sequence index. Edge lengths are also shown near the edges. See also Fig. 7 for several examples of the face sequences corresponding to the nodes above.

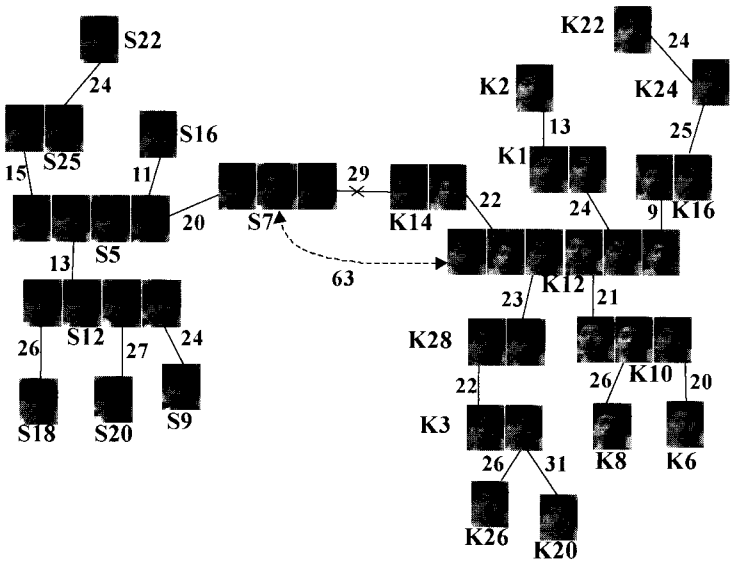


Fig. 3. In the initial MST graph, part of which is shown here, the face clusters for two subjects, subject S and subject K, are connected together by a spurious association link between nodes S7 and K14. The split procedure breaks that link after finding that $F(\mathrm{S7}, \mathrm{K12}, \mathrm{K14})$ in Eq. (6) has a positive value and applying Eqs. (7)–(10).

because it is computationally inexpensive and performs reasonably well in our case, where the faces can be easily normalized and each frame sequence contains a multitude of face templates including gradual variation in position, viewing angle, illumination, etc. More elaborate face distance measures might be used if processing time is not a problem, potentially leading to better recognition results. Conversely, if used in relation to different face-metric spaces, the present method can provide information about their efficacy, even in the cases when category-specific information might not be readily available.

From the distance matrices $\boldsymbol{M}^{(a,b)}$ is calculated the proximity matrix $\boldsymbol{P}$, whose elements $\rho\{a,b\}$ define the minimal distance between any two face sequences $F^{(a)}$ and $F^{(b)}$:

$$\rho\{a,b\} = \min_{x,y} dist\langle F^{(a)}(i,j,x), F^{(b)}(i,j,y)\rangle = \min_{x,y} m^{(a,b)}(x,y), \tag{2}$$

so that each face sequence is represented by a row in the symmetric $\boldsymbol{P}$. Here also, alternative between-sequence distance measures can be used (e.g. some variant of the Hausdorff distance, etc.). To each image sequence $F^{(a)}$, $F^{(b)}, \ldots, F^{(N)}$ is assigned a "node" $A, B, \ldots, N$, to represent it in a graph $\mathbf{G}$, constructed and updated by the learning algorithm (a real-data example is shown in Fig. 2). Using the proximity matrix $\boldsymbol{P}$, the initial set of nodes $\mathbf{C} = \{A, B, \ldots, N\}$ is divided into subsets $\mathbf{C_1}, \mathbf{C_2}, \ldots, \mathbf{C}_L$ ($L < N/2$) by connecting each node A by an edge $e(A,B)$ to a node B for which

$$B = \arg \min_{\substack{X \in \mathbf{C}, \\ X \neq A}} \rho\{A,X\}, \tag{3}$$

and grouping together in the same subset all nodes between which exists a path. The length (or weight) of an edge is equal to the distance between the two nodes, i.e. $|e(A,B)| = \rho(A,B)$. After each node is connected to its nearest neighbor (determined by Eq. (3)), depending on the data in $\boldsymbol{P}$, initial "*chains of associations*" will be formed, i.e. the disjoint sets $\mathbf{C_1}, \mathbf{C_2}, \ldots, \mathbf{C}_L$, each of which contains a chain of similar views linked by edges across the face sequences. Next, each set $\mathbf{C}_i$ is connected to its nearest-neighboring set $\mathbf{C}_j$ by an edge $e(A_i, B_j)$ between the pair of nodes

$$(A_i, B_j) = \arg \min_{\substack{X \in C_i, \\ Y \in C_j}} \rho(X,Y), \tag{4}$$

thus forming a new set $\mathbf{C}_{ij} = \mathbf{C}_i \cup \mathbf{C}_j$. This procedure of connecting nearest-neighboring sets and merging their elements is repeated recursively until all sets are merged into one final set, which is identical with the initial set $\mathbf{C} = \{A, B, \ldots, N\}$. Together, the set of nodes $\mathbf{C}$, the set $\mathbf{E}$ of the $N-1$ edges connecting the nodes in $\mathbf{C}$, the rule $\boldsymbol{\Gamma}$ for node connectivity (based on the nearest-neighbor principle), and the distance function ρ, define a *graph* $\mathbf{G}(\mathbf{C}, \mathbf{E}, \boldsymbol{\Gamma}, \rho)$, which by construction is a *tree* (i.e. a connected graph with no cycles, Ref. [18]). Also, since the sum of the weights of the edges of $\mathbf{G}(\mathbf{C}, \mathbf{E}, \boldsymbol{\Gamma}, \rho)$ is minimal, it is known as a *minimal spanning tree* (MST). MSTs have been proposed for detecting and describing Gestalt clusters by Zahn [19], and here we extend this approach to image sequences, although with several modifications (introduced in the following sections) which try to resolve some of the problems accompanying this method.

After the initial MST graph is formed, for the purpose of discrimination between different categories of faces, the edges $e(A,B) \in \mathbf{E}$, connecting two image sequences A and B in $\mathbf{C}$, are divided into two subsets $\mathbf{E}^+$ and $\mathbf{E}^-$ ($\mathbf{E} = \mathbf{E}^+ \cup \mathbf{E}^-$), so that "consistent" edges $e^+(A,B) \in \mathbf{E}^+$ are said to *connect* nodes which belong to the same face category (same person), and "inconsistent" edges $e^-(A,B) \in \mathbf{E}^-$ to *separate* nodes belonging to different categories, i.e. the latter designate the boundary between two different categories. Initially, all edges are labeled as consistent, and after that each edge $e(A,B) \in \mathbf{E}$ in $\mathbf{G}$ is re-assigned a new consistency label, based on the following consistency rule:

Consistency rule Φ: The consistency of the edge $e(A,B)$ between any two nodes A and B in $\mathbf{G}$ is determined by the following binary function $\phi(A,B)$, which assigns consistency labels "1: *consistent*" or "0: *inconsistent*" to each $e(A,B)$ in $\mathbf{G}$:

$$\phi(A,B) = \begin{cases} 1: \left(\rho(A,B) < \sigma_1 \dfrac{\sum_{A_i \in \mathbf{A}-\{B\}} \rho(A,A_i)}{\xi(\mathbf{A})-1}\right) \\ \qquad \text{AND} \left(\rho(A,B) < \sigma_1 \dfrac{\sum_{B_i \in \mathbf{B}-\{A\}} \rho(B,B_i)}{\xi(\mathbf{B})-1}\right), \\ 0: \text{otherwise.} \end{cases} \tag{5}$$

In Eq. (5), $\mathbf{A}$ ($\mathbf{B}$) is the set of all nodes $A_i(B_i)$ connected by a consistent edge to node $A(B)$, $\xi(\mathbf{A})(\xi(\mathbf{B}))$ is its cardinality, and σ_1 is a constant, called a *factor of inconsistency*. If some node in the pair (A,B) is a terminal node, then the condition in which it participates is considered to be satisfied. The order of consistency label assignment is important: it starts with the largest edge and proceeds in descending order of the edge sizes. The graph obtained after the consistency label assignment (5) is performed for each edge, is defined by $\mathbf{G}(\mathbf{C}, \mathbf{E}^+ \cup \mathbf{E}^-, \boldsymbol{\Gamma} + \boldsymbol{\Phi}, \rho)$, where the rule for consistency label assignment $\boldsymbol{\Phi}$ in Eq. (5) is added to the rule $\boldsymbol{\Gamma}$ for node connectivity. The inconsistent edges $e(A,B) \in \mathbf{E}^-$ partition $\mathbf{C}$ into disjoint sets (clusters) $\mathbf{C}_i$ ($i: 1, \ldots, K$), and K is the number of different face categories obtained by the construction of $\mathbf{G}$, i.e. in $\mathbf{G}$ all nodes connected by consistent edges are considered to belong to the same face category, while inconsistent edges partition sample space into clusters belonging to different categories. In this way, any two image sequences linked together by a consistent edge are said to form a "*consistent association*", and the structure of each cluster obtained by traversing the consistent edges in the resultant *chain of associations*, is completely defined

by the subgraph $\mathbf{G}_i(\mathbf{C}_i \subseteq \mathbf{C}, \mathbf{E}_i^+ \subseteq \mathbf{E}^+, \mathbf{\Gamma} + \mathbf{\Phi}, \rho) \subseteq \mathbf{G}$. Fig. 2 shows an example of the resulting graph using real data samples, some additional samples of which are shown also in Fig. 7.

2.3. Splits and merges of associations

MSTs can represent clusters with arbitrary forms which is very useful in the case of faces or other real-world 3D objects whose patterns' distributions vary significantly with changes of view point, illumination conditions, etc. However, in order for the MSTs to function properly, the assumption that the distances between similar views of the same object are *always* sufficiently shorter than the distances between similar views of different objects needs to be satisfied. In reality, this assumption might not be satisfied for many reasons: noise, imperfect features or distance functions, etc. As data accumulates, the probability of occurrence of pairs of associations from different categories satisfying the consistency constraint also increases. In such circumstances, the very useful property of the MST, to chain together similar associations, might turn into a defect of the method, resulting in the so-called *chaining effect* (Everritt [11]), which links together samples from different categories into a single cluster. In order to alleviate this problem, the graph $\mathbf{G}(\mathbf{C}, \mathbf{E}^+ \cup \mathbf{E}^-, \mathbf{\Gamma} + \mathbf{\Phi}, \rho)$ from Section 2.2 is further modified using the association split/merge procedures introduced in this section, which utilize local statistical information to identify and correct the inter-category boundaries overridden by the chaining effect.

2.3.1. Association splits

In the MST graph $\mathbf{G}$, for each non-terminal node X, which is connected by consistent edges to $\eta(X)$ ($\eta(X) \geqslant 2$) nodes (i.e. serves as a bridge connecting each pair among the $\eta(X)$ nodes), execute the following *association split* procedure.

Association split algorithm

For each of the $\eta(X)(\eta(X) - 1)/2$ different association pairs represented in $\mathbf{G}$ by nodes Y_j and Y_k ($j,k : 1 \ldots \eta(X);\ j \neq k$) calculate the binary functions

$$F(Y_j, Y_k, X) = \begin{cases} 1 : (Y_j, Y_k) > \sigma_2 \dfrac{1}{\eta(X)} \displaystyle\sum_{i=1}^{\eta(X)} \rho(X, Y_i), \\ -1 : \text{otherwise} \end{cases} \tag{6}$$

where σ_2 is a constant ($\sigma_2 \geqslant \sigma_1$). A positive value for $F(Y_j, Y_k, X)$ implies that node X bridges two associations coming from different categories, and in order to avoid the resulting chaining effect, one or both of the consistent edges $e^+(Y_j, X)$ and $e^+(Y_k, X)$ have to be replaced by an inconsistent edge. Which edge will become inconsistent is determined by evaluating $F(Y_j, Y_i, X)$ and $F(Y_k, Y_i, X)$ for all values of i ($i : 1 \ldots \eta(X);\ i \neq j;\ i \neq k$), as follows:

if $(F(Y_j, Y_i, X) > 0$ AND $F(Y_k, Y_i, X) > 0)$ *for some* i

$\Rightarrow$ replace $e^+(Y_j, X)$, $e^+(Y_k, X)$ with $e^-(Y_j, X)$, $e^-(Y_k, X)$; (7)

if $(F(Y_j, Y_i, X) < 0$ AND $F(Y_k, Y_i, X) < 0)$ *for all* i

$\Rightarrow$ replace $e^+(Y_j, X)$, $e^+(Y_k, X)$ with $e^-(Y_j, X)$, $e^-(Y_k, X)$; (8)

if $(\{F(Y_j, Y_i, X) < 0$ *for all* $i\}$ AND $\{F(Y_k, Y_i, X) > 0$ *for some* $i\})$

$\Rightarrow$ replace $e^+(Y_k, X)$ with $e^-(Y_k, X)$; (9)

if $(\{F(Y_j, Y_i, X) > 0$ *for some* $i\}$ AND $\{F(Y_k, Y_i, X) < 0$ *for all* $i\})$

$\Rightarrow$ replace $e^+(Y_j, X)$ with $e^-(Y_j, X)$. (10)

In order to distinguish between the inconsistent edges obtained as a result of the rule (5) during the initial MST formation, and the inconsistent edges obtained by Eqs. (6)–(10), we will call the former *inconsistent edges of type* 1, and the latter *inconsistent edges of type* 2. Two associations connected by an inconsistent edge of type 2 are said to form a pair of *spurious* associations, because although the edge between them satisfies the consistency criterion $\mathbf{\Phi}$ in Eq. (5), they yield a positive value for $F(\cdot)$ in Eq. (6), i.e. they are considered to belong to different categories. In the split procedure above, only nodes at depth 1 (i.e. only the nodes directly connected by consistent edges to the bridge-node X) are considered in the calculations (6)–(10), but if necessary, nodes at arbitrary depth can be included, the extreme case being when all nodes in the current cluster are involved in the calculations for each bridge-node X. An example illustrating the split procedure applied to some real data is shown in Fig. 3.

2.3.2. Association merges

After the split procedure above is executed for all nodes $X \in \mathbf{G}$, it is possible that together with the spurious association links, some legitimate ones might also have been split. This over splitting is especially likely to occur in the initial stages of the learning process (when not enough image sequences are available), and as a result the real clusters might be broken down into several smaller clusters, typically into one or few larger and many much smaller clusters-satellites. This situation can be significantly improved if the association split procedure is followed by a recursive *association merge* procedure. Also, the association split procedure guarantees that no spurious association chains (as defined by Eq. (6)) exist in $\mathbf{G}$, but does not guarantee that the structure of the tree is optimal with regard to consistency. After the splitting, the graph $\mathbf{G}$ still has the same connectivity as the initial MST, only the type of its constituent edges has

changed. Therefore, $\mathbf{G}$ is still optimal in the sense that the sum of the lengths of its connecting edges is minimal. For our task, however, we would rather relax the condition for optimality in regard to the sum of edge lengths, in order to obtain optimality in the sense that the resulting tree structure would partition sample space into the least number of clusters (i.e. the number of inconsistent edges is minimal), while at the same allowing no spurious chains of associations to corrupt the discriminative power of the representation. In order to achieve this type of optimal tree structure we propose the following merge algorithm.

Merge algorithm

Step 1: Sort in ascending order all clusters obtained after the execution of the association split procedure, i.e. all clusters separated from each other by inconsistent edges type 1 or type 2.

Step 2: Attempt to merge each cluster $\mathbf{C}_K$ to the nearest cluster $\mathbf{C}_L$ by connecting with a consistent edge their respective nodes X_{C_K} and X_{C_L}, for which the following conditions are satisfied simultaneously:

$$(1)\ F(Y_j, Y_k, X_{C_K}) < 0 \quad \text{for all } j,k \quad (j,k : 1 \ldots \eta(X_{C_K});\ j \neq k) \tag{11}$$

$$(2)\ F(Y_j, Y_k, X_{C_L}) < 0 \quad \text{for all } j,k \quad (j,k : 1 \ldots \eta(X_{C_L});\ j \neq k) \tag{12}$$

$$(3)\ \xi(\mathbf{C}_K) \leqslant \xi(\mathbf{C}_L) \tag{13}$$

Step 3: Exit if no new cluster merges occur in step 2, otherwise continue with step 4.

Step 4: Sort all obtained clusters in ascending order and go to step 2.

In *step 2* above, $\xi(\mathbf{C})$ is the cardinality of cluster $\mathbf{C}$, and the functions $F(\cdot)$ are defined as in Eq. (6). If such nodes X_{C_K} and X_{C_L} exist, then the inconsistent edge between $\mathbf{C}_K$ and $\mathbf{C}_L$ will be deleted and substituted by the new consistent edge $e^+(X_{C_K}, X_{C_L})$. If more than one pairs of nodes (X_{C_K}, X_{C_L}) satisfy Eqs. (11)–(13), that pair the edge between which has the shortest length is chosen. If a pair of nodes satisfying Eqs. (11)–(13) does not exist, $\mathbf{C}_K$ and $\mathbf{C}_L$ are not merged. Requirements (1) and (2) in step 2 ensure that the newly inserted consistent edge $e^+(X_{C_K}, X_{C_L})$ will not connect $\mathbf{C}_K$ and $\mathbf{C}_L$ by a spurious associative link. Again, in order to distinguish between the consistent edges obtained as a result of the consistency rule $\mathbf{\Phi}$ in Eq. (5) during the initial MST formation, and the consistent edges inserted during the merge step, the former are named *consistent edges of type* 1, and the latter *consistent edges of type* 2. The process of modifying the initial MST by the split and merge procedures is illustrated in a schematic form in Fig. 4(a)–(c). The modified tree structure, obtained after the execution of the association split/merge procedures above, is used to determine the final clustering. It can be symbolically represented as $\mathbf{G}(\mathbf{C}, \mathbf{E}^+ \cup \mathbf{E}^-, \mathbf{\Gamma} + \mathbf{\Phi} + \mathbf{\Psi}, \rho)$, where $\mathbf{\Psi}$ reflects the process of substitution of *consistent edges* in $\mathbf{G}(\mathbf{C}, \mathbf{E}^+ \cup \mathbf{E}^-, \mathbf{\Gamma} + \mathbf{\Phi}, \rho)$ with *inconsistent edges type* 2 during the associations splits, and replacement of *inconsistent edges* with *consistent edges type* 2 during the associations merges.

2.4. Incremental learning and online recognition

This section describes the incremental version of the algorithm introduced in the previous two sections. When a new image sequence is available from the input, its corresponding node can be easily added to an already existing graph $\mathbf{G}$ (which initially might consist of only two nodes) in incremental fashion. First, assume that the current internal state of the system is represented by the relations between $N (N > 1)$ nodes in $\mathbf{G}$, to which the newly available $(N+1)$st node has to be added. This can be easily accomplished by the following algorithm for incremental node addition.

Algorithm for incremental node addition

Step 1: In the proximity matrix $\boldsymbol{P}$ compute the $(N+1)$st row corresponding to the new node X_{N+1}.

Step 2: Find the *nearest-non-spurious neighbor* of X_{N+1}, node K (see Fig. 5a):

$$K = \arg \min_{Y \in \mathbf{A}(X)} \rho(Y, X_{N+1}). \tag{14}$$

In Eq. (14), $\mathbf{A}(X)$ is the set of all nodes X which if connected to X_{N+1} by a consistent edge $e^+(X, X_{N+1})$, at least one of the following three conditions would be satisfied:

$$1.\ F(X_{N+1}, Y_i, X) < 0 \quad \text{for all } Y_i,\ (i : 1 \ldots \eta(X)); \tag{15}$$

$$2.\ F(X_{N+1}, Y_i, X) < 0 \quad \text{for all } Y_i, \text{ for which } \rho(X_{N+1}, X) < \rho(X, Y_i); \tag{16}$$

$$3.\ \rho(X_{N+1}, X) < \min_i \rho(X, Y_i), \tag{17}$$

where the functions $F(\cdot)$ are defined in Eq. (6), Y_i are all nodes connected by a consistent edge to X, and $\eta(X)$ is their number. Condition 1 in Eq. (15) is satisfied by all nodes X which would form a non-spurious association link if connected to X_{N+1}. Condition 2 in Eq. (16) is satisfied by all X which would form a non-spurious association link when connected to X_{N+1}, if the edges $e(X, Y_i)$ to neighbors farther away than X_{N+1} are ignored. If $\mathbf{A}(X) = \{\varnothing\}$, connect X_{N+1} to its nearest neighbor by an inconsistent edge and jump to step 4. Otherwise, connect nodes K and X_{N+1} by a consistent edge $e^+(K, X_{N+1})$ and replace all $e^+(K, Y_i)$ for which

$$\rho(Y_i, K) > \sigma_1 \frac{1}{\eta(K)+1} \left[\sum_{i=1}^{\eta(K)} \rho(K, Y_i) + \rho(K, X_{N+1}) \right] \tag{18}$$

with $e^-(K, Y_i)$. After that, in case Eq. (15) was not satisfied for node K, all $e^+(K, Y_i)$ for which $F(X_{N+1}, Y_i, K) > 0$ are set inconsistent.

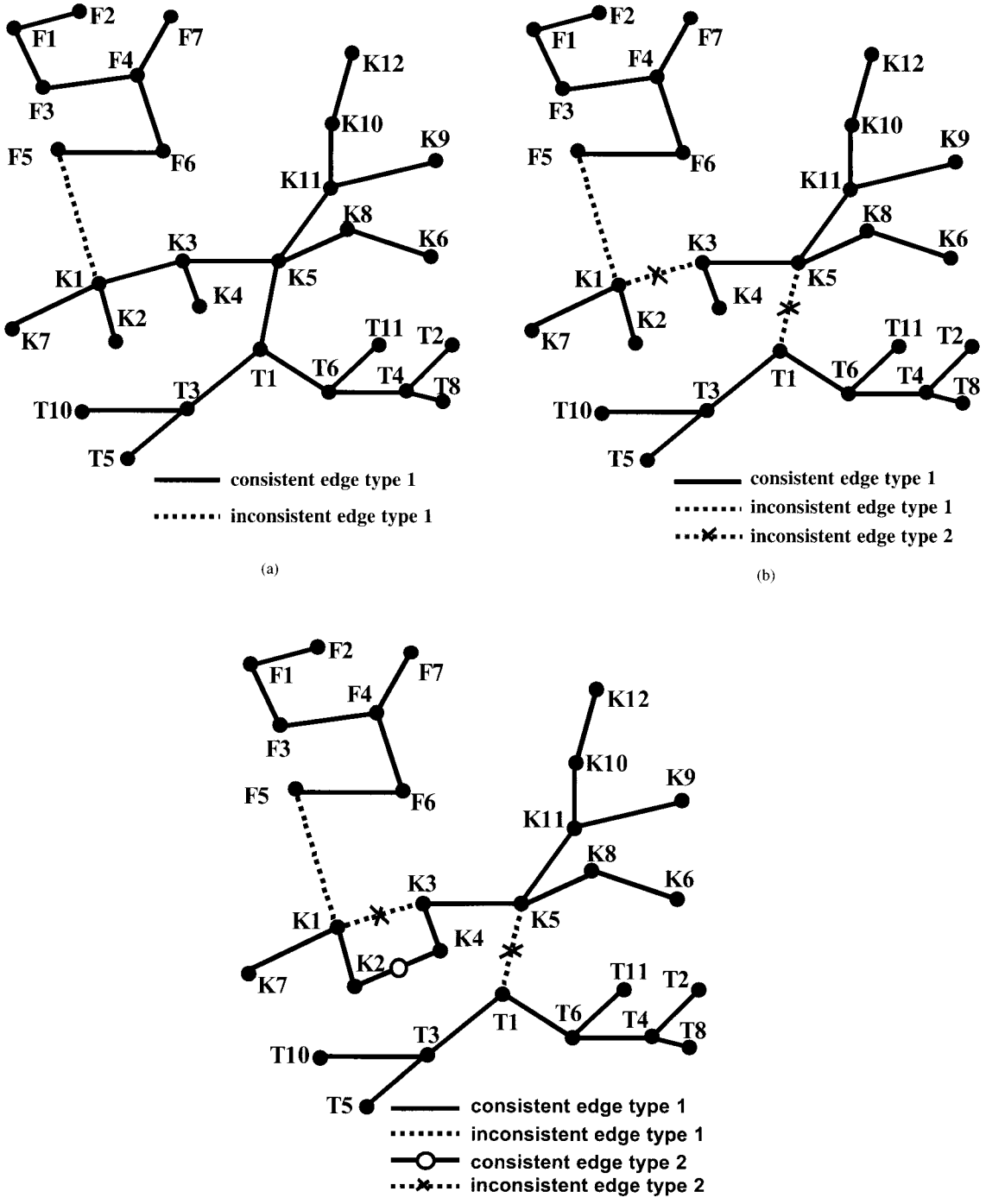


Fig. 4. Schematic illustration of the association split/merge procedure. The initially built MST graph is shown in (a), where two clusters are found (subjects K and T are grouped together because of the *chaining* effect). In (b), two spurious edges are marked as inconsistent type 2 by the split procedure, as a result of which the cluster for subjects K and T is split successfully, but at the same time the cluster for subject K is also split into two at nodes Kl and K3. In (c), the merge procedure reconnects the two groups for subject K by inserting a consistent edge of type 2 between nodes K2 and K4.

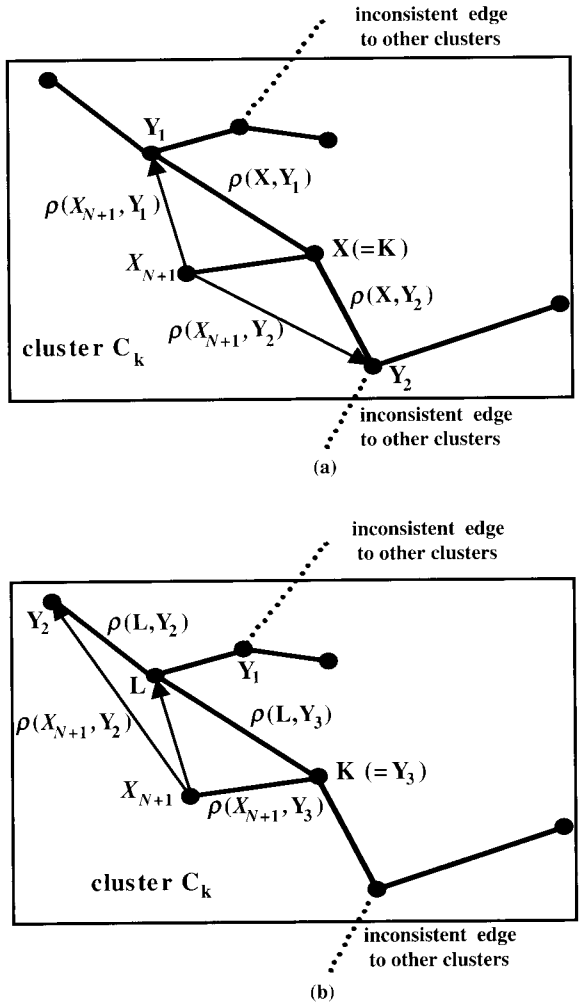


Fig. 5. Incremental node addition—steps 2 and 3. Intra-cluster edges are updated after the addition of the new node X_{N+1}: (a) illustration for step 2; (b) illustration for step 3. See text for details.

Step 3: After the necessary node insertions/deletions in step 2 are accomplished, for all nodes L belonging to the same cluster as node $K (L, K \in \mathbf{C}_K)$, use the distances $\rho(L, Y_i)$ between L and all nodes Y_i connected directly to L by a consistent edge, to reorganize $\mathbf{G}$ in the following way (see Fig. 5b). First, let the subgraph $\mathbf{G}_K (\mathbf{G}_K \subset \mathbf{G})$ representing cluster $\mathbf{C}_K$ be divided into two separate sub-trees, $\mathbf{G}_K^1$ and $\mathbf{G}_K^2$, if $e^+(L, Y_i)$ is temporarily removed from $\mathbf{G}_K$. Then consider the following two cases:

1. If removal of $e^+(L, Y_i)$ would place Y_i and the new node X_{N+1} in the same sub-tree (i.e. $Y_i, X_{N-1} \in \mathbf{G}_K^1 \Rightarrow L \in \mathbf{G}_K^2$, or $Y_i, X_{N+1} \in \mathbf{G}_K^2 \Rightarrow L \in \mathbf{G}_K^1$), then if $\rho(L, X_{N-1}) < \rho(L, Y_i)$, and $e^+(L, X_{N+1})$ would not form a spurious association link, delete $e^+(L, Y_i)$ and insert a new consistent edge between L and X_{N-1}.
2. If removal of $e^+(L, Y_i)$ would place Y_i and the new node X_{N+1} in different sub-trees (i.e. $L, X_{N+1} \in \mathbf{G}_K^1 \Rightarrow Y_i \in \mathbf{G}_K^2$, or $L, X_{N+1} \in \mathbf{G}_K^2 \Rightarrow Y_i \in \mathbf{G}_K^1$), then if $\rho(X_{N+1}, Y_i) < \rho(L, Y_i)$, and $e^+(X_{N+1}, Y_i)$ would not form a spurious association link, delete $e^-(L, Y_i)$ and insert a new consistent edge between X_{N+1} and Y_i.

It is easy to show that the above edge insertions/deletions would update $\mathbf{G}$, preserving its tree structure optimal in the sense explained in Section 2.3.2. Steps 2 and 3 in the algorithm for incremental node addition correspond to the MST formation and association split steps in the batch version of the AC algorithm, while step 4 below corresponds to the association merge step.

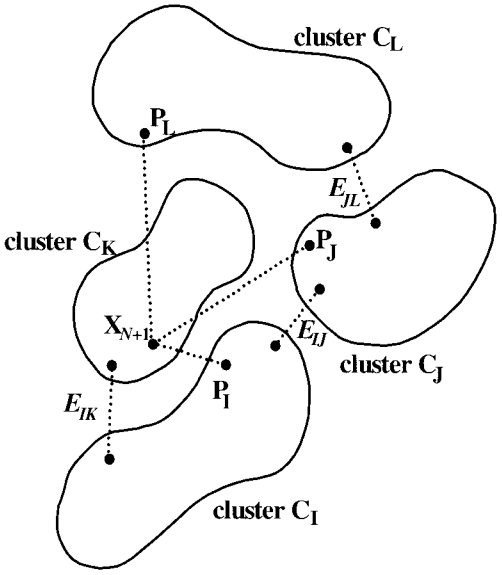
Step 4: In steps 2 and 3 above, the intra-cluster edges in $\mathbf{C}_K$ were updated, and now the same thing has to be done for the inter-cluster (inconsistent) edges of $\mathbf{G}$, to reflect the influence of the addition of the new node X_{N-1}. Let $\mathbf{C}_K$ be the cluster to which the new node belongs (see Fig. 6), i.e. $X_{N-1} \in \mathbf{C}_K$, and $\mathbf{C}_I$ and $\mathbf{C}_J$ are any two clusters in $\mathbf{G}$ connected by an inconsistent edge E_{IJ} with length $|E_{IJ}|$. For each $\mathbf{C}_I$ in $\mathbf{G}$, use the distances $\rho(X_{N-1}, P_I)$, where

$$P_I = \arg\min_{Y_a} \rho(X_{N+1}, Y_a), \quad a : 1 \ldots \xi(\mathbf{C}_I), \tag{19}$$

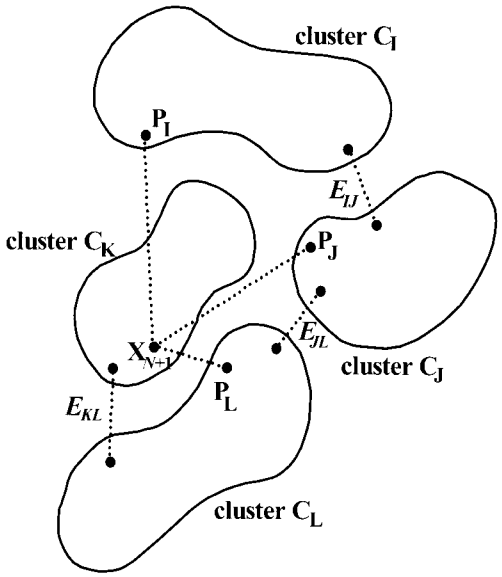
is the nearest node to X_{N+1} among all nodes $Y_a \in \mathbf{C}_I$, to reorganize $\mathbf{G}$ in the following way. For all $\mathbf{C}_I$ in $\mathbf{G}$ (including also the case $\mathbf{C}_I \equiv \mathbf{C}_K$), consider the temporal removal of the inconsistent edge E_{IJ} to each of its neighboring clusters $\mathbf{C}_J$ ($\mathbf{C}_J \neq \mathbf{C}_K$), which would divide $\mathbf{G}$ into two separate sub-trees, $\mathbf{G}^1 \subset \mathbf{G}$ and $\mathbf{G}^2 \subset \mathbf{G}$, $\mathbf{G}^1 \cap \mathbf{G}^2 = \{\varnothing\}$. Then the following two cases are considered:

1. If removal of E_{IJ} would place $\mathbf{C}_K$ and $\mathbf{C}_J$ in different sub-trees (see Fig. 6a), i.e. $\mathbf{C}_K \subset \mathbf{G}^1 \Rightarrow \mathbf{C}_J \subset \mathbf{G}^2$, or $\mathbf{C}_K \subset \mathbf{G}^2 \Rightarrow \mathbf{C}_J \subset \mathbf{G}^1$, then if $\rho(X_{N+1}, P_J) < |E_{IJ}|$, delete E_{IJ} and insert a new edge between P_J and X_{N+1}. The consistency of $e(P_J, X_{N+1})$ is determined depending on whether it would or would not form a spurious association link.
2. If removal of E_{IJ} would place $\mathbf{C}_K$ and $\mathbf{C}_J$ in the same sub-tree (see Fig. 6b), i.e. $\mathbf{C}_K \subset \mathbf{G}^1 \Rightarrow \mathbf{C}_J \subset \mathbf{G}^1$, or $\mathbf{C}_K \subset \mathbf{G}^2 \Rightarrow \mathbf{C}_J \subset \mathbf{G}^2$, then if $\rho(X_{N+1}, P_I) < |E_{IJ}|$, delete E_{IJ} and insert a new edge between P_I and X_{N-1}. The consistency of $e(P_I, X_{N+1})$ is determined depending on whether it would or would not form a spurious association link.

The learning algorithm explained in this section can start with some data gathered in advance, which is processed in an offline "batch" manner, while subsequent additions of new data are executed incrementally in an online manner. Alternatively, it is possible to ensure online incremental performance from the very beginning. If only two nodes (face sequences) are available initially, they are connected by a consistent edge, and further incoming input data is added



(a)



(b)

Fig. 6. Incremental node addition—step 4. Inter-cluster edges are updated after the addition of the new node X_{N+1}. See text for details.

sequentially, node by node, using the algorithm for incremental node addition described above.

Online recognition/verification can be implemented identically to the node addition algorithm. A node corresponding to the test sequence is inserted in the most recent version of the graph **G** using the node addition algorithm from this section, and the category of the test sample is determined to be the same as the one of the cluster to which it is connected by a consistent edge. In case the test sample is connected by an inconsistent edge, it is rejected as a face which has not been learnt yet.

3. Experimental results

In order to evaluate the performance of the proposed method, several experiments have been conducted using more than 300 face image sequences obtained over a period of several months from 17 different subjects. A typical example of the experimental setting can be seen in Fig. 1, and several time-subsampled face sequences for different people, together with time stamp labels obtained from the preprocessor, can be seen on Fig. 7. The illumination conditions were demanding and varied significantly with the time of the day during which the samples were taken. The video sequences' length varied between 30 and 300 frames, depending on the speed with which the subjects walked in front of the camera, in the range between slow walking with occasional stops, and running. Between 7 and 40 sequences were gathered for each subject. Two different data sets were

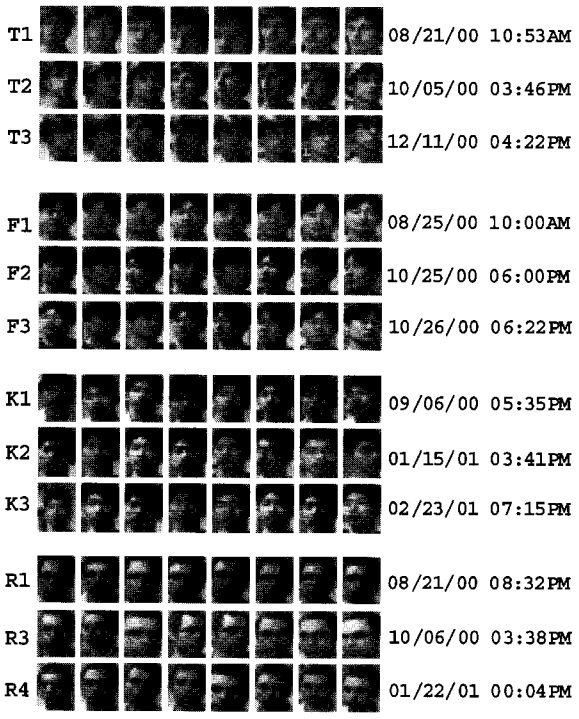


Fig. 7. An example of several time subsampled face sequences, with category labels (to be obtained by the algorithm) shown to the left, and the time stamps labels available from the preprocessor, shown to the right.

Table 1
Experimental results

Data set	Sequences used	ε_{AB}	ε_0	R (%)
A	200	11	3	93.0
B	177	6	19	85.9
$A+B$	377	9	34	88.6

used in the experiments below:

(a) *Data set A*: In this data set, the subjects were just walking forward toward the camera. Predominantly frontal faces were included, with a few side-view faces at the end of the sequences, when the subjects passed beside the camera. Sequences T1, F1, K1, K3, R1, R4 in Fig. 7 are representative for the data included in this set;

(b) *Data set B*: In this data set, the subjects were told to look to the left and right, up and down, as they moved towards the camera. Both frontal and side-view faces were represented in this data set. Sequences T2, T3, F2, F3, K2, R3 in Fig. 7 are representative for the data included in this set.

Samples with and without glasses were included for all subjects who had such, and hair length/styles changed with time. Resolution of the original images was 320×240 pixels, and 18×22 pixels for the normalized face-only images. Near real-time processing was achieved on a SGI O2 workstation with a R12000 (300 MHz) processor. The following formula was used for calculating the recognition (self-organization) rate R:

$$R=\left(1.0-\frac{\varepsilon_{AB}+\varepsilon_0}{N}\right)\times 100\%, \tag{20}$$

where N is the total number of sequences to be grouped, ε_{AB} is the number of sequences which are mistakenly grouped into the cluster for certain category A, although in reality they come from category B, and ε_0 is the number of samples gathered in clusters in which no single category occupies more than 50% of the nodes inside them. The following three experiments were conducted, with results given in Table 1. In all experiments data from all 17 subjects were used.

Experiment 1 Only data from data set A were used, where predominantly frontal faces were included.

Experiment 2 Only data from data set B were used, where both frontal and side-view face images were included.

Experiment 3 Both data sets A and B (all available data) were used.

The results obtained in experiments 1–3 above indicate that the AC algorithm could provide a simple and efficient way to detect (in an unsupervised manner) and represent category groups of complex 3D objects, such as faces, whose appearance might vary significantly with changes in view angle, illumination conditions, etc. This is made possible by the formation of association links involving different views in the individual sequences, as for example can be seen in Fig. 2 for the sequences F4, R3, R5. It should be noted that the tree **G** obtained as a result of the AC algorithm is functionally very different from the trees obtained in standard hierarchical clustering algorithms, in which every node represents a single individual pattern. In **G**, each node, although designated by a single point in the graph, might represent several very different patterns. For example, if a certain node X in **G** is directly connected to nodes A and B, it does not mean that the face connected to A and the face connected to B are identical, or even similar. Actually, one of these faces may be a frontal face and the other a side-view face, the distance between which could be very large in terms of the concrete metric being used. To further illustrate this point using a somewhat more extreme example, suppose that for a certain subject the following two sequences are available: one containing only frontal face views (e.g. about 0°), and one containing only side views (e.g. near 90°), as those shown in the top and bottom rows of Fig. 8. Even with more sophisticated features or distance measures than those used in our experiments, it would be difficult to recognize that these face sequences belong to the same subject. In the AC algorithm, rather than relating *directly* those two sequences to each other, or to a certain reference center (like the centroids in the *k-means*-variant algorithms, which would put severe restrictions on the structural form of the clusters), the clustering is performed by relating each one of them to a third sequence (e.g. as the one shown in the center row of Fig. 8), or a connected group of sequences, which by virtue of the gradual sensory variation displayed in them, provide the links necessary to discover the relation between

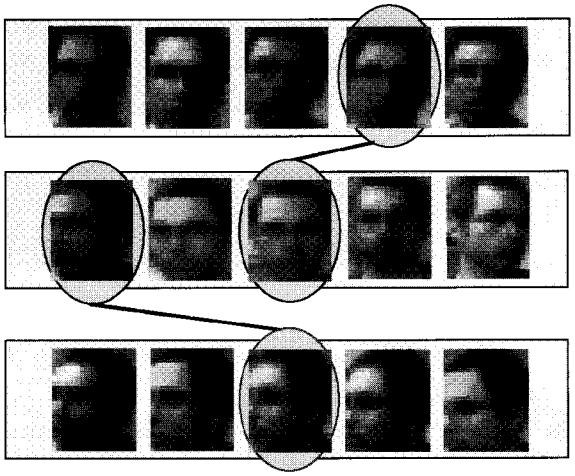


Fig. 8. Illustration of the mechanism by which two very different views of the same subject (with high inter-sequence dissimilarity value) can be grouped together using the AC algorithm. The sequences in the top and bottom row contain only frontal and side views, respectively. Consistent associations formed between each of these sequences and the face sequence in the center row (which contains both frontal and side views of the same person) makes it possible to label the three sequences with the same category.

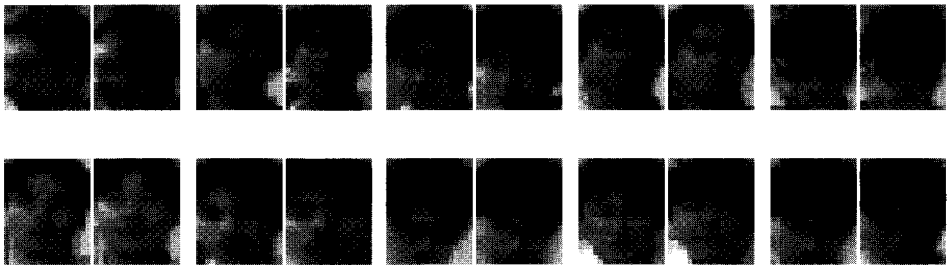


Fig. 9. Several examples of pairs of faces from different categories, found to be very similar (nearest neighbors) by the face matching procedure (1).

the seemingly distant samples. As a result of this, traversing the "chain" of consistent associations eventually permits to group together samples which might seem very far from each other in terms of the available metrics. It should be noted that this strategy is applicable not only to the case of different views, but also to differences in illumination conditions, scale, resolution, etc., and even can be applied to other sensory modalities.

Regarding the clustering errors in the experimental results reported in Table 1, part of them can be explained with the limitations inherent in the face matching technique (1) used to determine the distance between two face images (see Fig. 9 for several examples illustrating this problem). Using more sophisticated face matching schemes, or calculating the inter-sequence distances in a better selected feature space would further improve recognition. Also, because of the large volume of input data which had to be processed, it was impossible to check whether all face images in each of the resultant image sequences obtained from the preprocessor were legitimate ones, and this also might have contributed to some of the clustering errors. Some automatic procedure for face validation has to be included in the future versions of the system. The relatively lower recognition rates obtained in the more difficult experiment 2 can be explained to some extent with insufficient data—typically less than one face sequence per week was available for most subjects. Additionally, part of the data samples in set B contained excessive and exaggerated head movements, which would be unlikely to happen in real situations, but were included as a more difficult test. The overall performance improved when set B was mixed with set A in Experiment 3. In conclusion, despite the above-mentioned problems, the experimental results still can be considered promising, having in mind the difficulty of the task and the demanding environmental conditions in which they were obtained.

4. Conclusion

In this paper we have proposed a novel method for unsupervised face recognition from video sequences of time-varying face images obtained over an extended period of time in real-world conditions. The learning process implemented by the method does not rely on category-specific information provided by human teachers in advance (which might be biased by their limited understanding of the complex real-world environment), but rather lets the system find out by itself the structure and underlying relations inherent in the sensory input. The proposed method provides the following important advantages: (a) it allows all stages of the resulting face recognition system to be completely automated, avoiding the need for manual segmentation and labeling of the input stream. Manual segmentation and labeling of the input stream might be impractical and sometimes impossible, e.g. in online video surveillance systems; (b) this permits to train the system with a sufficient quantity of input data, providing the higher level of sensory variation necessary for such a challenging task as the one attempted here; (c) both frontal and side view faces can be learnt/recognized by the method; (d) the proposed method has a natural incremental implementation, allowing for "non-destructive" learning, which also may be important in online systems dealing with large databases.

Results from several experiments using both frontal and side-view face sequences obtained under demanding illumination conditions were reported here, achieving recognition rate of 88.6% for the data set obtained until now. Although the preliminary results are encouraging (having in mind the difficulty of the task), additional tests with much larger data sets have to be done in order to obtain further insights about the limitations and possibilities of the present method.

Acknowledgements

The authors are grateful to Dr. K. Ishii and Dr. N. Hagita of NTT Communication Science Laboratories for their help and encouragement.

References

[1] A. Samal, P.A. Iyengar, Automatic recognition and analysis of human faces and facial expressions: a survey, Pattern Recognition 25 (1992) 65–77.

[2] R. Chellapa, C.L. Wilson, S. Sirohey, Human and machine recognition of faces: a survey, Proc. IEEE 83 (1995) 705–740.
[3] M.A. Grudin, On internal representation in face recognition systems, Pattern Recognition 33 (2000) 1161–1177.
[4] H. Wechsler, P.J. Philips, V. Bruce, F.F. Soulie, T.S. Huang (Eds.), Face Recognition: From Theory to Applications, Springer, Berlin, 1998.
[5] H. Ando, S. Suzuki, T. Fujita, Unsupervised visual learning of three-dimensional objects using a modular network architecture, Neural Networks 12 (1999) 1037–1053.
[6] J.J. Weng, W.S. Hwang, Toward automation of learning: the state self-organization problem for a face recognizer, Proceedings of the Third International Conference on Automatic Face and Gesture Recognition, 1998, pp. 384–389.
[7] D.L. Swets, J. Weng, Hierarchical discriminant analysis for image retrieval, IEEE Trans. PAMI 21 (5) (1999) 386–401.
[8] S.M. Omohundro, Best-first model merging for dynamic learning and recognition, in: J.E. Moody, S.J. Hanson, R.P. Lippmann (Eds.), Advances in Neural Information Processing Systems, Vol. 4, Morgan Kaufmann Publishers, San Mateo, CA, 1992, pp. 958–965.
[9] S. Satoh, Comparative evaluation of face sequence matching for content-based video access, Proceedings of the Fourth International Conference on Automatic Face and Gesture Recognition, 2000, pp. 163–168.
[10] A.K. Jain, R.C. Dubes, Algorithms for Clustering Data, Prentice-Hall, Englewood Cliffs, NJ, 1988.
[11] B.S. Everitt, Cluster Analysis, Wiley, New York, 1993.
[12] R. Duda, P. Hart, D. Stork, Pattern Classification, Wiley, New York, 2001.
[13] H.A. Rowley, S. Baluja, T. Kanade, Neural network based face detection, IEEE Trans. Pattern Anal. Mach. Intell. 20 (1) (1998) 23–38.
[14] T. Darrell, G. Gordon, J. Woodfill, M. Harville, A virtual mirror interface using real-time robust face tracking, Proceedings of the Third International Conference on Automatic Face and Gesture Recognition, 1998, pp. 616–621.
[15] E. Hjelmas, B.K. Low, Face detection: a survey, Comput. Vision. Image Understand. 83 (2001) 236–274.
[16] P. Viola, M. Jones, Rapid object detection using a boosted cascade of simple features, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Kauai, Vol. I, 2001, pp. 511–518.
[17] T. Sim, R. Sukthankar, M. Mullin, S. Baluja, Memory-based face recognition for visitor identification, Proceedings of Fourth International Conference on Automatic Face and Gesture Recognition, 2000, pp. 138–142.
[18] M. Gondran, M. Minoux, Graphs and Algorithms, Wiley, New York, 1990, pp. 129–141.
[19] C.T. Zahn, Graph theoretic methods for detecting and describing Gestalt clusters, IEEE Trans. Comput. C-20 (1) (1971) 68–86.

About the Author—BISSER RAYTCHEV received his B.S. and M.S. degrees in Electronics from Tokai University, Japan, and his Ph.D. in Electronics and Information Sciences from Tsukuba University, Japan. He is currently a research associate at NTT Communication Science Labs, NTT Corporation. His current research interests include biological and computer vision, pattern recognition and machine learning.

About the Author—HIROSHI MURASE received the B.E., M.E., and Ph.D. degrees in Electrical Engineering from Nagoya University, Japan. From 1980 to the present he has been engaged in pattern recognition research at Nippon Telegraph and Telephone Corporation (NTT). From 1992 to 1993 he was a visiting research scientist at Columbia University, New York. He was awarded the Telecom System Award in 1992, the IEEE CVPR best paper award in 1994, the IEEE ICRA best video award in 1996, and the Takayanagi Award in 2001. His research interests include computer vision, video analysis, character recognition, and multimedia information recognition. He is a member of the IEEE, and the Information Processing Society of Japan.